Visualization of Audio in Two and Three Dimensions

Zachary Hoegberg

October 21, 2016

1 Abstract

My project will consist of two parts: a program to interpret audio signals, i.e. music, in a visual way, and an interactive program to create audio from a visual file. Audio files are essentially an array that returns amplitude as a function of time. Therefore, my project will make use of the forward and inverse Fourier transforms in order to turn the time-dependent functions into an array of amplitude and phase shift for every frequency.

2 Theory

2.1 Fourier Transform

Humans hear sound by converting pressure waves in the surrounding air into electrical signals. Microphones work essentially the same way, and speakers work by taking that electrical signal and moving air to create pressure waves. For this reason, we can think of an audio file as a function of time which returns the pressure of the sound wave. However, since a sound wave is a periodic function with a varying amplitude, we can decompose it into a sum of sine waves, each with a particular amplitude and phase shift. This can be accomplished by using the Fourier transform:

$$\hat{f}(\xi) = \int_{-\infty}^{+\infty} f(x) e^{-2\pi i x \xi} dx$$

where ξ is frequency in Hz and x is time in seconds.

Given a function $\hat{f}(\xi)$, we can also turn it back into a function of time by using the inverse Fourier transform:

$$f(x) = \int_{-\infty}^{+\infty} \hat{f}(\xi) e^{2\pi i \xi x} d\xi$$

This allows us to take a user-created function or graph of frequencies and turn it into a function that can be used to generate an audio file.

2.2 Fast Fourier Transform

Since the Fourier transform works on a continuous function and our input is a set of discrete points, we need a slightly different algorithm to compute the answer:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-2\pi ikn/N}, k \in 0 \leq \mathbb{Z} \leq N-1$$

where N is the length of the input array containing the values of x_n. This returns a complex number for each frequency from $k = 0$ up to $50Hz$, which can be turned into a trigonometric function using Euler's formula. Using this, we can compute a new set of forward and inverse discrete Fourier transform:

$$X_k = \sum_{n=0}^{N-1} x_n (\cos(-2\pi k \frac{n}{N}) + i \sin(-2\pi k \frac{n}{N}))$$

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k (\cos(2\pi k \frac{n}{N}) + i \sin(2\pi k \frac{n}{N}))$$

These equations are the basis of an effecient algorithm for quickly computing the Fourier transform of a data set called the Fast Fourier Transform.

3 Project

3.1 Part 1: Visualization

The first part of the program involves taking the Fourier transform of a user-supplied audio file and displaying several graphs related to it. First, a simple graph of the pressure over time will be displayed. Next, a display of the Fourier transform of the entire file will be displayed. Finally, an animated graph will be displayed that will show the Fourier transform of a small period of the file as the file is being played.

3.2 Part 2: Creating a Sound File

This part will allow a user to "draw" a picture consisting of multiple points representing different frequencies and amplitudes. By creating an array of the points and performing an inverse Fourier transform, this array can be turned into an array that can be saved and played back as a sound file.

4 Checkpoints and Timeline

4.1 Part 1

1. Many modules exist to perform Fourier anlysis already. I have to take the results of the transform and filter out the strong frequencies from the noise. I expect to have this working by the end of October.

2. Creating a function that can take the Fourier transform of a smaller array in real time will be the next step. I should be able to use the same algorithm; most of the work here will be determining the best size of the array and optimizing the code so it can run in real time. I expect to have this done before Thanksgiving break.

3. Creating the visualizations will be the last step. I essentially want to create three graphs, each showing a different array and one of them changing with time. I will have this completed by the end of Thanksgiving break.

4.2 Part 2

1. The first step in this program will be creating an algorithm to take an array, run the inverse Fourier transform, and create a sound file. This could still be used even if the graphing part of this program is not completed. I would like to have this done by the end of the first week of December.

2. The second step of this program will be creating a user interface to enable graphing the desired points and turning them into an array. I hope to have this completed before finals week.